



This shows that VS has completed all of the necessary work to make sure that our web services can be used in this UI project.

Next, open the `showall.aspx.cs` file; we will change the code to display a list of customers who can use the web service interface. as follows:

```
/// <summary>
/// Get and Show a list of all customers
/// </summary>
private void FillAllCustomers()
{
    localhost.OMSService service =
    new NTier.Web.localhost.OMSService();
    CustomerDTO[] customers = service.GetAllCustomers();
    rptCustomers.DataSource = customers;
    rptCustomers.DataBind();
}
```

In the above method, we create a new instance of our `OMSService` and call it as a service. Then we call the `GetAllCustomers` web method that returns an array of customer data transfer objects. We then bind these DTOs to the repeater. Note that in this code, we did not use any object in the domain model directly (as was done in the same code in Chapter 4). We used only the service interface and the data transfer object that belongs to the `NTier.Common` project, which is common to all the projects.

Now let us understand how we can use this service in an external project. First, we need to host this web service on a server so that it can be discovered and referenced by external programs. For this, simply create a virtual directory in your local web server (you can name it **OMSService**) and then publish the `NTier.Service` project to that virtual directory. We will create a simple console application using Visual Studio (VS) and then call this web service to display a list of customers. Here are the steps:

1. Create a new VS solution and select **Add New Project**.
2. Select **Console Application** and name it **Service Consumer**.
3. Right-click on the newly-added project in VS and select **Add Reference**.
4. Add a reference to the **5Tier.Common** project. This is needed because we need to use the `CustomerDTO` class, which is being returned by our service. If we don't want to use the `5Tier.Common` project, then we can create our own DTO class with the same name and having the same properties.
5. Right-click the project again and select the **Add Service Reference** option.
6. In the dialog box that is displayed, type the address of the OMS web service, which you hosted on your local server. The address should look something like this: `http://localhost/OMSService/OMSService.asmx`. Here is a screenshot of how the dialog box will look:

